Slovenská technická univerzita v Bratislave

FAKULTA INFORMATIKY A INFORMAČNÝCH TECHNOLÓGIÍ

Ilkovičova 3, 842 16 Bratislava 4

Ponuka

Analýza textov pracovných ponúk z webu

Tímový projekt Inžinierskeho štúdia

Tím č. 5 :

Bc. Blanárik Ivan
Bc. Kočiš Ladislav
Bc. Kotrba Attila
Bc. Kročka Lukáš
Bc. Lenický Mário

Október, 2005

Obsah

0 Úvod

Mail: tpteam5@pobox.sk

0.1 Slovo na úvod

Žijeme v spoločnosti, kde informácie zohrávajú veľkú úlohu v živote človeka. Vzdelanie a informovanosť sú dôležitou podmienkou pri výbere nášho povolania. V súčasnosti je najväčším zdrojom informácií Internet. V tejto sieti sú informácie uložené v takej podobe, aby boli ľahko spracovateľné počítačmi. Internetové prehliadače prezentujú tieto informácie z Internetových stránok do podoby čitateľnej pre človeka. Samotné Internetové stránky však obsahujú množstvo irelevantných informácií, ktoré nie sú pre človeka dôležité.

Častou požiadavkou mnohých informačných systémov je aktuálnosť informácií a možnosť spolupráce s inými systémami. Preto je dôležité pravidelne sledovať zmeny, ktoré v týchto systémoch nastali. Manuálne sledovanie týchto zmien zbytočne zaťažuje človeka, ktorý je touto úlohou poverený, a pri rozsiahlych systémoch ani nie je možné. Preto vznikla požiadavka na automatizované sledovanie takýchto zmien – na automatizovanú aktualizáciu informácií. Nástrojom na realizáciu tejto požiadavky je obaľovač (wrapper). Informácie získané takýmto obaľovačom je potom možné použiť ako vstup do iných systémov (napríklad na zobrazenie a vyhľadávanie takýchto informácií na mobilných telefónnych zariadeniach – prostredníctvom služby WAP).

Praktické využitie takéhoto obaľovača je napríklad na získavanie informácií o pracovných ponukách z Internetových stránok pracovných agentúr. Výstupom budú informácie v štruktúrovanej podobe, ktoré sú vhodné pre ďalšie spracovanie.

Cieľom tohto dokumentu je ponúknuť riešenie takéhoto obaľovača, ktoré máme v úmysle riešiť v rámci predmetu Tvorba softvérového systému v tíme. Toto riešenie obaľovača by sme chceli aj prakticky využiť na sprostredkovanie a zobrazenie takto získaných informácií o pracovných ponukách priamo na našich mobiloch prostredníctvom služby WAP.

0.2 Zadanie projektu

Obaľovač (angl. obaľovač) je program, ktorý slúži na získavanie informácií z webových stránok. Jeho použitie ušetrí manuálnu prácu sledovania informácií na webových stránkach (resp. informácií zadaných v inom formáte). Cieľom projektu je identifikácia a stiahnutie relevantných informácií z neštrukturovaného kódu HTML, ktorá je zameraná najmä na prezentáciu informácií pre človeka a nie na spracovateľnosť pre počítače.

Existujú obaľovače predprogramované na špecifickú stránku. Adaptívne obaľovače pracujúce pre väčšiu sadu stránok, ktoré však nemusia identifikovať všetky požadované informácie pre každú stránku. Zoberme napríklad stránky pre pracovné ponuky. Pracovné ponuky sú publikované na rôznych stránkach v rôznych zápisoch. Zvyčajne obsahujú podobné informácie (miesto práce, pozícia, plat, atď.) Ak uvažujeme o implementácii predprogramovaných obaľovačov, implementujeme obaľovač pre každú stránku zvlášť a očakávame, že obaľovače stiahnu všetky informácie uvedené na stránkach korektne. Ak sa rozhodneme implementovať adaptívny obaľovač, očakávame, že nájde informácie o pracovných ponukách z ľubovoľnej stránky (resp. väčšej množiny stránok), ktorá takéto informácie obsahuje. Adaptívny obaľovač môže prehľadávať web, identifikovať webové stránky a ukladať nájdené detaily o pracovných ponukách. Má však oveľa horšiu úspešnosť než predprogramovaný obaľovač.

V tomto projekte sa očakáva implementácia predprogramovaného obaľovača. Pri tvorbe obaľovačov narazíme na niekoľko zaujímavých problémov, ktoré bude treba riešiť:

- parsovanie HTML dokumentov – webové stránky sú zvyčajne nekorektné. Pre prekonanie tohto problému sa používajú parsre, ktoré pracujú aj s nekorektným HTML kódom (mozilla, beutifulsoap, atď.) alebo sa kód najskôr prekonvertuje do korektného XHTML pomocou nejakej implementácie htmltidy.
- verifikácia korektnosti obaľovača – nakoľko stránky sa môžu časom meniť, buď kvôli zmene rozhrania alebo kvôli meniacim sa HTML kódom, ktoré sú zadané manuálne ľuďmi, treba pre obaľovače implementovať kontrolný mechanizmus. Ak obaľovač nezvláda stránku, na ktorú je prispôsobený, pri kontrole korektnosti údajov vývojár obaľovača dostane upozornenie.
- čitateľnosť a rozšíriteľnosť obaľovača – obaľovač má byť rozšíriteľný a dobre čitateľný kód. Treba preto implementovať framework a dodržiavať istú kultúru pri jej implementácii. Implementácia obaľovača deklaratívnym jazykom môže zľahčiť udžiavateľnosť kódu.
- navigácia – často najväčší problém je dostať sa stránku, ktorá obsahuje informácie, ktoré potrebujeme. Po ceste môžu byť 'cookies', heslá, 'sessions'.

Od tímu sa očakáva adresovanie týchto problémov, implementácia jednoduchého softvérového nástroja a vypracovanie štýlu implementácie obaľovačov. Riešenie má byť demonštrované v doméne pracovných ponúk.

0.3 Predstavenie členov tímu

Bc.Lukáš Kročka

Má odborné vedomosti a skúsenosti s programovacími jazykmi C++ (aj práca s knižnicou MFC), Java, PHP a HTML. Skúsenosti s jazykmi C, Power Builder a databázou Oracle si upevňuje prácou popri štúdiu v spoločnosti Soluziona Slovakia. Témou bakalárskeho projektu boli Multimediálne informácie v regionálnom IS, kde sa zdokonalil v jazykoch PHP, Java, HTML, v používaní CSS štýlov a databázou MySQL. Na projektoch v škole nadobudol skúsenosti aj s CASE nástrojom Rational Rose (UML), ktorý slúži na modelovanie softvérových systémov vo všetkých fázach životného cyklu. Znalosti z jazyka XML má možnosť nadobudnúť aj pri diplomovom projekte. Rád by nadobudol vedomosti z jazyka C#, ktorý sa stáva jeden z najpoužívanejších webových technológií, alebo skúsenosti s tvorbou WAP stránok.

Bc. Ladislav Kočiš

Má bohaté skúsenosti s programovacími jazykmi C a C++. Počas štúdia získal znalosti prostredia Visual C++ a jeho knižnice MFC. Dlhodobo sa zaujíma o prostredie .NET Framework, hlavne o jazyk C# a súvisiace technológie ASP .NET a ADO .NET. Náplňou jeho bakalárskeho projektu bol systém na odovzdávanie zadaní prostredníctvom internetu. V tomto projekte si upevnil znalosti C# a celého prostredia .NET Framework. Takisto ovláda jazyky HTML, CSS a JAVASCRIPT. Pracuje vo firme Ditec a.s., kde si upevňuje a rozširuje znalosti prostredia .NET a jazyka C#. V projektoch pracuje na tvorbe databázovej, business a z časti aj prezentačnej vrstvy.Ovláda prácu s jazykom XML a databázou MS SQL. Takisto využijeme aj jeho predchádzajúce skúsenosti s tvorbou obaľovača.

Bc. Ivan Blanárik

Bakalárske štúdium ukončil na FIIT STU Bratislava v študijnom odbore softvérové inžinierstvo. Venuje sa programovaniu v jazykoch C a C++ v grafickom prostredí Windows využívaním Win32API a MFC. Má skúsenosti s prácou v jazykoch Java a HTML a i.. Počas štúdia získal znalosti z oblasti softvérového inžinierstva a používania UML.

Bc. Attila Kotrba

V bakalárskej práci sa venoval abstraktným výpočtovým strojom a oblasti teoretickej informatiky. Praktické skúsenosti s prácou v tíme nadobudol počas pracovnej praxe popri štúdiu. Dlhodobo sa venuje práci s platformou .NET, a to konkrétne programovaciemu jazyku C# a ďalším nadväzujúcim technológiám, ako ADO.NET, Webové služby. Má skúsenosti s prácou s databázami Oracle a MS SQL, s vytváraním databázových skriptov, prácou na webových projektoch technológiou ASP.NET. Pracoval na projektoch rôzneho rozsahu, v súčasnosti pracuje na komplexnom softvérovom systéme pre tretí pilier dôchodkových poisťovní, kde pracuje hlavne na vývoji databázovej a business vrstvy aplikácií. Z ďalších znalostí je to jazyk C++ a modelovanie v nástrojoch ako Enterprise Architect a Rational Rose. Jeho súčasným záujmom je jazyk UML a návrhové vzory, taktiež sa venuje novým technológiám.

Bc. Mário Lenický

Náplňou jeho bakalárskeho projektu bolo automatizované vizuálne zobrazenie inventára firmy v budovách na viacerých podlažiach. Aplikačná časť bola typu klient – server (prispôsobená web rozhraniu) a samotná sada vizualizačných skriptov bola naprogramovaná v interpretačnom jazyku PHP. Databázová časť bola založená na MySQL. Pracoval na viacerých web projektoch stredného rozsahu (rozpočet 20 000 – 200 000 Sk), kde uplatnil skúsenosti s jazykmi HTML, PHP a SQL. Disponuje skúsenosťami s programovacím nástrojom Microsoft Visual C++ a knižnicou MFC; Vypracoval win32 aplikáciu stredného rozsahu pre stravovaciu firmu. Popri štúdiu nadobudol prax a pokročilé skúsenosti s prácou v tíme v spoločnosti Siemens PSE, kde v päťčlennom tíme spolupracoval na vývoji diagnostického parsera výstupov z mobilných ústrední. Pri vývoji tejto aplikácie si rozšíril obzor v oblasti MFC a CASE nástrojov, pričom sa zdokonalil vo fáze špecifikácie požiadaviek a objektovo orientovaného návrhu softvéru.

0.4 Motivácia

Táto téma nás zaujala, pretože v nej vidíme možnosť praktického využitia v praxi. Skúsenosti, ktoré by sme nadobudli pri riešení tohto zadania, nám môžu byť veľmi užitočné pri výbere budúceho povolania resp. pri riešení ďalších projektov na našej fakulte.

Ako kvalifikovaný tím môžeme ponúknuť riešenie, ktoré sme navrhli na základe širokej škály vedomostí a znalostí z oblasti tvorby softvérových systémov. Tieto sme nadobudli počas štúdia na našej fakulte a aj z praxe. Náš tím má skúsenosti v oblasti webových technológií ako sú C#, PHP, MySQL, CSS štýly a samozrejme HTML. Vo viacerých diplomových projektoch v súčasnosti riešených členmi nášho tímu figuruje aplikácia znalostí XML. Navyše v prípade potreby členovia tímu dokážu rýchlo a flexibilne zvládnuť nové technológie potrebné k vývoju požadovaného softvéru.

Garantujeme vytvorenie obaľovača s jednoduchou možnosťou údržby v prípade zmeny štruktúry zdrojových kódov, ktoré budú vstupom tohto obaľovača. V prípade, že obaľovač nebude schopný získať požadované informácie z týchto zdrojových kódov, upozorní vývojára a ten sa rýchlo postará o jeho opravu. Obaľovač bude sťahovať zdrojové súbory priamo z Internetu. Z týchto zdrojových súborov sa odstránia všetky nadbytočné informácie, ktoré nie sú zapísané v tvare HTML (napríklad rôzne komentáre, Javascripty a pod.).
Podrobnejšie informácie o návrhu sú uvedené v kapitole 2.2 Návrh obaľovača.

Prehľadný formát výstupu bez nadbytočných dát, nízke nároky na údržbu a možnosť nasadenia obaľovača pre zobrazenie jeho výstupných dát prostredníctvom služby WAP predurčujú náš produkt pre klienta, ktorý chce po zavedení produktu do činnosti racionálne využiť finančné zdroje potrebné na údržbu dodaného softvéru a zaškolenie personálu. Zavedenie tohto produktu do prevádzky by mohlo prispieť

k čiastočnému zníženiu nezamestnanosti vďaka dostupnosti mobilných telefónov. Ďalej by mohlo slúžiť pre študentov na rýchle a pohodlné vyhľadávanie brigádnických činností popri štúdiu.

1 Prehľad problémovej oblasti

1.1 Prehľad problémového prostredia

Pri hľadaní informácií o pracovných príležitostiach je nutné prezerať veľké množstvo stránok, na ktorých sú uverejňované pracovné ponuky. Manuálne otváranie všetkých stránok a následné preklikanie sa k požadovaným informáciám však môže zaberať značné množstvo času.

Z dôvodu uľahčenia tejto činnosti sa používa tzv. obaľovač (wrapper), ktorý z webových stránok získa potrebné informácie a po spracovaní ich zobrazí na obrazovku (alebo takto získané údaje z obaľovača môžeme použiť ako vstup pre iné systémy – viď spomínané mobilné telefóny).

Základnou a najdôležitejšou funkciou Obaľovača ako takého je výstup pracovných ponúk. Vo všeobecnosti existujú dva druhy obaľovačov, a to obaľovače

- Adaptívne – Tieto dokážu prechádzať rozmanité stránky, hľadajú kľúčové slová, a z nich vytvárajú výstupy. Tieto obaľovače majú nízku úspešnosť, pretože musia prekonávať neznáme prekážky pri hľadaní stránok a ich parsovaní, keďže sa nevie ani ich približný tvar.
- Predprogramované – Tieto prechádzajú vopred zadefinované stránky, ktorých štruktúra je známa, a preto sú informácie z nich získané kvalitnejšie a dosahujú oveľa lepšiu úspešnosť. Ich nevýhodou však je, že pri zmene štruktúry danej stránky sa musí zmeniť aj obaľovač.

Obaľovače sa musia vedieť vysporiadať s mnohými problémami. Niektoré web stránky sú chránené prístupovým menom a heslom. Prvým problémom, ktorý musia obaľovače riešiť, je prechod cez tieto prístupové web stránky na web stránky s požadovanými údajmi.

Každá web stránka je zapísaná v html kóde. Výstupom väčšiny web stránok je nekorektný html kód. Môžu tam byť chyby, ako napr. vynechanie koncových tagov, prekríženie tagov, atď. Všetky tieto chyby by nemali ovplyvniť prácu obaľovača. Po nejakom čase môžu web stránky zmeniť svoju štruktúru do tej miery, ktorá môže ovplyvniť správnosť sťahovania dát obaľovačom. Obaľovač by mal byť schopný tieto zmeny detekovať a v čo najskoršom termíne oznámiť autorovi obaľovača.

Ďalším problémom je samotná navigácia medzi web stránkami. Mnohé web stránky používajú tzv. cookies a sessions na uchovávanie svojho stavu a identifikáciu používateľa počas prechodov medzi stránkami. Obaľovač musi byť schopný pamätať si cookies aj sessions.

1.2 Ciele produktu

Primárnym cieľom finálneho produktu je maximalizácia korektnosti výstupu a abstrakcie údajov, aby sa dali použiť ako vstup pre generovanie WAP stránky a následne zobraziť priamo na displeji mobilného telefónu.

Ďalším z cieľov je jednoduchosť údržby a obsluhy, čím sa značne minimalizujú časové a finančné zdroje klienta potrebné po dodaní požadovaného softvérového produktu.

Dôležitým aspektom obaľovača je miera záťaže na hardvérové zdroje, ktorá je výrazným faktorom ovplyvňujúcim výkon a stabilitu obaľovača. Naším cieľom bude minimalizovať záťaž najmä na procesor a pamäť stanice, na ktorej bude zabezpečený beh obaľovača, či už to bude klientská alebo serverová stanica. Docielime to efektívnym využitím pamäte a implementáciou jednoduchých a rýchlych algoritmov s nízkou výpočtovou zložitosťou.

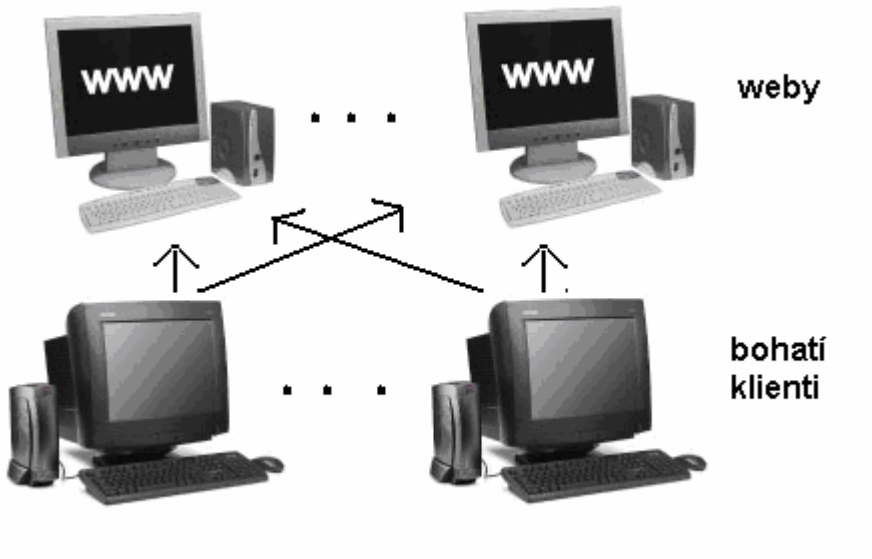
Z hľadiska softvérových požiadaviek na produkt cieľom nášho tímu je maximalizovať kompatibilitu dodaného systému. Keďže výber operačného systému značne ovplyvní výkon a stabilitu obaľovača, popri preferenciách klienta, ktoré budú zohľadňované s vyššou prioritou, je nutné dbať na výber čo možno najvhodnejšieho pracovného softvéru.

2 Návrh riešenia

2.1 Návrh architektúry

V tejto časti analyzujeme rôzne možnosti návrhu architektúry. Pre každú možnosť uvedieme jej výhody a nevýhody. Na konci kapitoly sa rozhodneme pre najvhodnejšiu možnosť.

2.1.1 Zaťažený klient



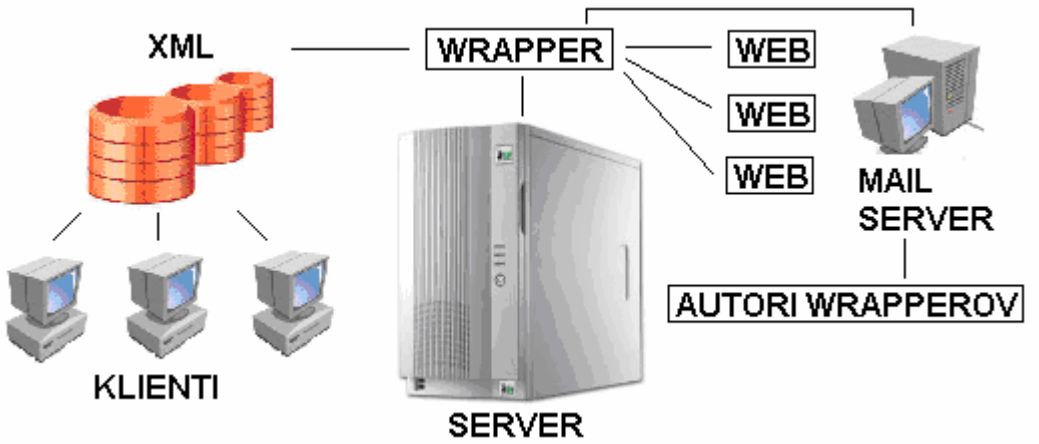
Obr. 1. Základná štruktúra zaťaženého klienta

Tento návrh architektúry je veľmi jednoduchý. Na obrázku vidíme weby, ktoré obsahujú ponuky pracovných príležitostí. Ďalej tam vidíme klientov, v tomto prípade sú to programy, ktoré sťahujú pracovné ponuky. Užívateľ si na klientovi vyberie kritéria pre vyhľadávanie pracovných ponúk. Obaľovač dá vyhľadávať ponuky zo všetkých webov, ktoré sú nastavené. Po vyhľadaní ponúk, klient zobrazí nájdené ponuky. Hlavným problémom tohto riešenia je, že klienti (predpokladáme veľké množstvo klientov) veľmi zaťažujú webové servery, čoho dôsledkom môže byť zablokovanie prístupového konta alebo zablokovanie ip adresy.

Výhody:
- jednoduchá architektúra

Nevýhody:
- nutnosť sťahovať klientský program
- klient beží len pod operačným systémom Windows (vid. Požiadavky na softvér)
- pri viacerých klientoch sa veľmi zaťažujú webové serveri, lebo dáta sa sťahujú pre každého klienta zvlášť
- nemožnosť automatického upovedomenia autora daného obaľovača, lebo klienti nemajú (veľká väčšina) SMTP server
- samotné vyhľadávanie prebieha na webových serveroch
- nedá sa použiť na generovanie wap stránok, je nutné použiť server

2.1.2 Zaťažený server typu I



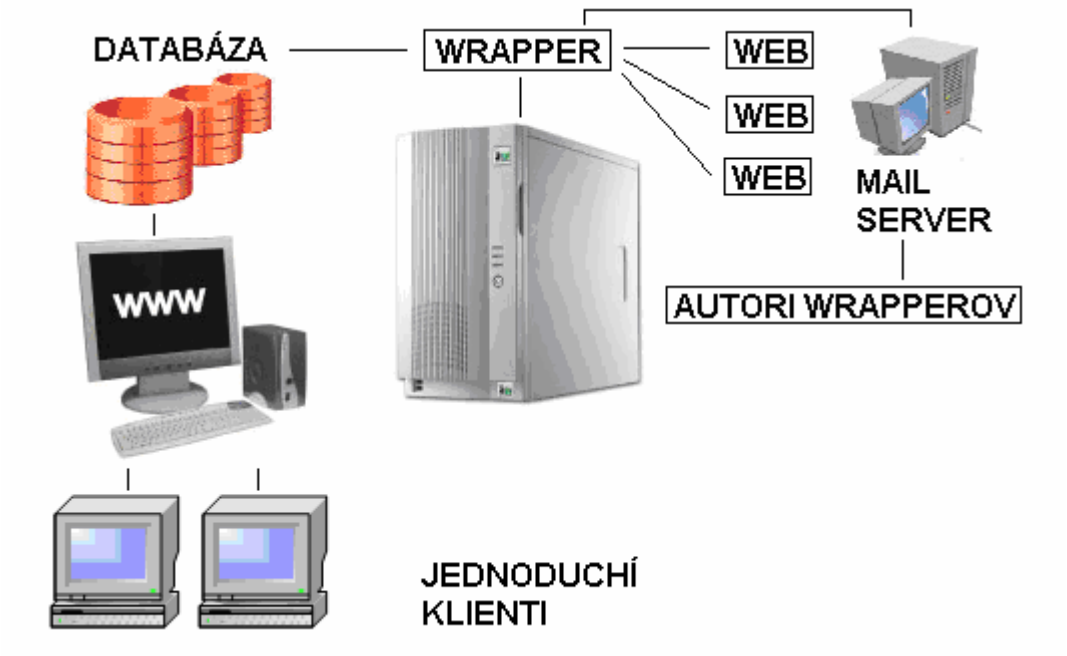
Obr. 2. Základná štruktúra zaťaženého servera typu I

Pri tomto návrhu architektúry sme preniesli obaľovač z klienta na server. Keďže každý(dobrý) server obsahuje poštový server, je možné automaticky posielať oznamy chýb príslušným autorom obaľovačov. Obaľovač sťahuje všetky pracovné ponuky v pravidelných intervaloch(max. raz za deň) a ukladá ich do xml súboru, ktorý je taktiež uložený na serveri. Príslušný klienti si stiahnu všetky ponuky zo servera. Klient potom vyhľadá používateľovi pracovné ponuky podľa zadaných kritérií.

Výhody:
- vyhľadávanie prebieha na klientovi
- menšia záťaž webu (sťahovanie pracovných ponúk sa uskutočňuje v nočných hodinách)
- jednoduchá architektúra
- možnosť automatického upozorňovania autorov obaľovačov pri výskyte chýb
- možnosť použitia pre generovanie wap stránok

Nevýhody:
- nutnosť sťahovať klienta
- klient beží len pod operačným systémom Windows

2.1.3 Zaťažený server typu II



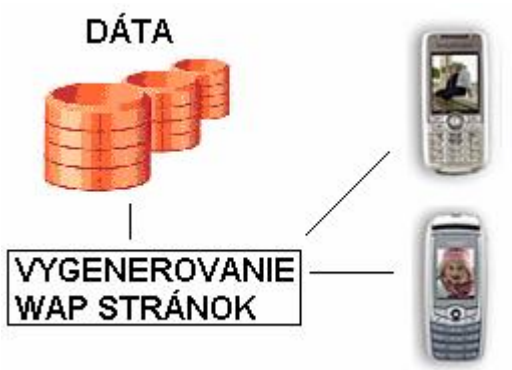
Obr. 3. Základná štruktúra zaťaženého servera typu II

V tomto návrhu sme vymenili úložisko dát. Stiahnuté pracovné ponuky sa budú uchovávať v databáze, namiesto v xml súbore. Dosiahneme tým vyššiu rýchlosť, hlavne pri vyhľadávaní pracovných ponúk podľa zadaných kritérií používateľa. Ďalej sme pridali web, cez ktorý sa bude pristupovať k pracovným ponukám. Ďalšou výhodou je, že používatelia môžu pristupovať na web z ľubovoľného miesta so svojim obľúbeným internetovým prehliadačom.

Výhody:
- menšia záťaž webu (sťahovanie pracovných ponúk sa uskutočňuje v nočných hodinách)
- možnosť automatického upozorňovania autorov obaľovačov pri výskyte chýb
- dostupnosť
- možnosť použitia pre generovanie wap stránok

Nevýhody:
- vyhľadávanie prebieha na serveri
- zložitejšia architektúra

2.1.4 WAP (možné rozšírenie predchádzajúcich dvoch návrhov)



Obr. 4. Základná štruktúra WAP

Tento návrh je len možným rozšírením predchádzajúcich dvoch architektúr. Do návrhu pridáme blok, ktorý bude generovať wap stránky zo stiahnutých pracovných ponúk.

Výhody:

- rozšírenie okruhu používateľov

Zhrnutie:

Uviedli sme tu možnosti, podľa ktorých by sa dal projekt riešiť. Najvýhodnejšia architektúra sa nám vidí tretia uvedená, teda zaťažený server typu II. Je veľmi zaujímavá, lebo sa dá rozšíriť o generovanie wap stránok a sprístupniť svoj obsah pre mobilné zariadenia. Je dostupná každému používateľovi. Avšak konečný výber by sme uskutočnili po podrobnejšej analýze. Treba zohľadniť aj možnosti využitia a časovú náročnosť jednotlivých riešení.

2.2 Navrhované implementačné prostredie

Pre implementáciu samotného obaľovača by sme si vybrali prostredie .NET Framework a jazyk C#. Jazyk C# bol navrhnutý priamo pre platformu .NET Framework. Je odvodený od jazykov C a C++. Je to jednoduchý, moderný a objektovo orientovaný jazyk. Je v mnohom podobný jazyku Java. Prostredie .NET Framework nám ponúka množstvo funkcií, ktoré nám výrazne uľahčia implementáciu tohto projektu. C# podporuje aj tvorbu wap stránok pre mobilné zariadenia, čo je ďalším dôvodom pre jeho výber. Rozhodli sme sa preň aj preto, lebo viacerí členovia nášho týmu majú praktické skúsenosti a znalosti tohto prostredia. [1]

2.3 Návrh Obaľovača

My budeme realizovať podľa zadania obaľovač predprogramovaný, teda webové stránky budú vopred známe. Požiadavka zo strany zadania bola, aby bol obaľovač ľahko modifikovateľný pre rôzne webové stránky, tomuto bol prispôsobený aj návrh.
Keďže údaje na web stránkach (pracovné ponuky) sa menia často, ale nepravidelne, obaľovač možno realizovať viacerými spôsobmi, a to :

A, Ako klasický spúšťateľný súbor, ktorý si v prípade potreby aktuálnych hodnôt bude môcť používateľ spustiť sám

B, Ako naplánovaný proces alebo službu operačného systému, ktorý sa bude spúšťať v pravidelných intervaloch, takže budú zabezpečené vždy aktuálne ponuky zo stránky

C, Ako webová služba, ktorú si môže zavolať používateľ, a jej volaním sa vygeneruje výstup, ktorý bude zaslaný používateľovi.

Všetky tieto spôsoby ale v zásade vyžadujú rovnakú funkcionality obaľovača, a teda budú realizované jednotne, budú sa líšiť len v prístupe k obaľovaču. V priebehu analýzy a návrhu sa rozhodneme ktorý spôsob použijeme, alebo môžeme použiť viacero spôsobov.
Základná štruktúra obaľovača sa skladá z nasledovných blokov:

A, Control

Blok sa sústreďuje na riadenie celej činnosti obaľovača. Keďže obaľovač je realizovaný predprogramovým spôsobom, prehľadávanie viacerých webových stránok sa realizuje spôsobom, že Control vyberá jednotlivé bloky Processing, ktoré sú implementované pre jednotlivé weby, a tým vyhľadáva pracovné ponuky na viacerých weboch.

B, Navigation

Základný problém pri obaľovači je dostať sa na stránku, na ktorej sú pracovné ponuky. Tento blok bude slúžiť na prechod medzi stránkami a dosiahnutie stránok s pracovnými ponukami. Tento blok musí prekonávať pri prístupe na stránku problémy ako prihlásenie sa na stránku, alebo použitie „cookies“ alebo „sessions“. Jeho výstupom je html stránka.

B, Processing

Ak je tento blok inicializovaný, prehľadáva konkrétny web, pre ktorý je implementovaný. Využíva pri tom blok Navigation, ktorý mu dodá html stránky, ktoré požaduje. Ak nájde stránku obsahujúcu pracovnú ponuku, túto prepošle ďalej do bloku Parser. Postupne tak prehľadáva celý web, pre ktorý bol naprogramovaný.

C, Parser

Vstupný dokument predstavuje html stránku. Tento dokument, ale nemusí byť v korektnej forme. Úlohou tohto bloku je z daného vstupného dokumentu získať text a atribúty pracovnej ponuky. Implicitne sa predpokadá, že na vstupnej stránke je len jedna pracovná ponuka. Pri svojej činnosti môže pre vyhľadávanie pracovnej ponuky na stránke použiť konverziu na korektný html kód, alebo tiež je tu možnosť použiť konverziu na Document Object Model, čo je všeobecný objektový model dokumentov aplikovateľný na rôzne typy dokumentov. Tento predstavuje na jazyku nezávislé rozhranie, ktorým sa môže pristupovať k dokumentom. Na vyhľadávanie možno tiež použiť aj regulárne výrazy. Výstup z tohto modulu predstavuje dokument, ktorý obsahuje „čisté“ pracovné ponuky.

D, Output

Tento blok zhromažďuje pracovné ponuky, formátuje a zoraďuje ich do výsledného dokumentu (pravdepodobne XML), ktoré po ukončení činnosti (našiel všetky pracovné ponuky) uloží do súboru na určené miesto, odkiaľ si ho môže zobraziť používateľ.

Tiež môže posielať upozorňujúce e-maily, napr. ak je nesprávny formát html stránky pri jej zmene.

E, Viewer

Súbor pracovných ponúk je bude štruktúrovaný dokument, ktorý je ale nevhodný pre priame pozeranie, a preto pre jeho zobrazenie je vhodnejšie použiť nejaké rozhranie, zobrazovač, napr. webstránku.

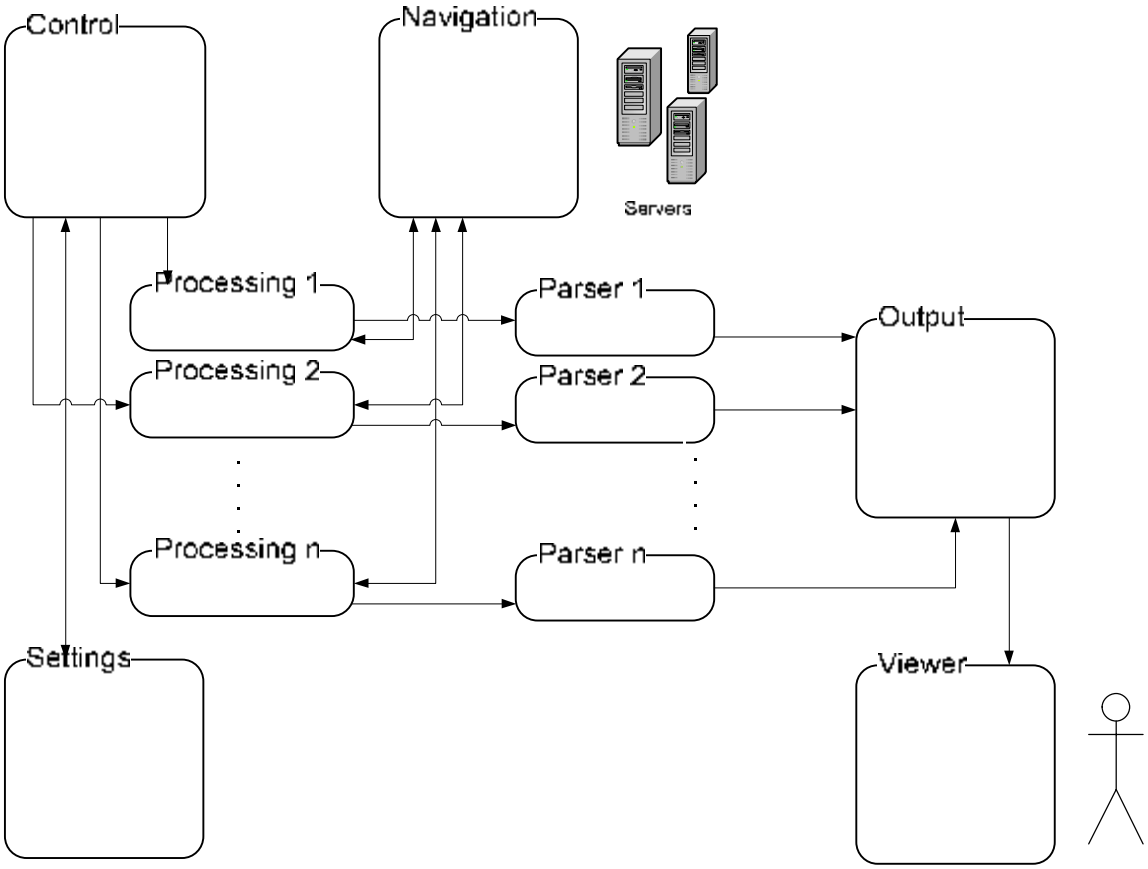
F, Settings

Celý obaľovač musí mať nastavenia uložené najlepšie na jednom mieste. Obaľovač má mnoho nastavení, ktoré budú v samostatnom bloku. Tiež tu môže byť zadefinovaný napr. formát výstupného súboru pracovných ponúk, alebo formát upozorňujúceho mailu.

Obaľovač teda bude realizovaný modulárnym spôsobom, jednotlivé horeuvedené bloky budú realizované ako samostatné moduly.

Medzi výhody tohto spôsobu patrí oddelený vývoj jednotlivých modulov, ktoré spolu budú komunikovať vopred určenými rozhraniami. Taktiež tieto bloky sú znovupoužiteľné aj v iných systémoch. V prípade zmeny stránky, alebo realizácie nového obaľovača pre novú stránku netreba meniť a kompilovať všetky moduly obaľovača, ale len niektoré, a to Processing a Parser. Ostatné moduly môžu ostať bez zmeny, a teda nemusia byť prekódované. Túto výhodu považujeme za kľúčovú, pretože v zadaní je požadovanou vlastnosťou obaľovača možnosť rýchleho prekódovania pri zmene stránky. Ak by sme chceli zmenu formátu výstupu, tak stačí zmeniť len modul Output. Toto splňuje aj požiadavku zadania, a to rýchlu zmenu obaľovača pri výmene stránky. Obaľovač realizovaný modulárne je možno s úpravami použiť aj na inú domému, ako doménu pracovných ponúk, v tomto prípade treba zmeniť moduly Processing a Parser, a tiež v prípade potreby aj zobrazovač.

Ako nevýhody tohto riešenia oproti riešeniu celého obaľovača v celku, „šitého“ na jednu jedinú stránku vidím dlhšie trvajúcu implementáciu, pretože treba jasne definovať čo patrí do ktorého modulu, a jednotlivé rozhrania medzi modulmi. Tiež celkový čas spracovávania môže byť o niečo dlhší, ale táto nevýhoda by sa nemala veľmi negatívne uplatniť.



Obr. 5. Základná bloková štruktúra obaľovača

2.4 Požiadavky

2.4.1 Požiadavky na hardvér

Minimálne požiadavky:

Klient: Pentium 300MHz alebo jeho ekvivalent, 128MB RAM

Server: Pentium 600MHz alebo jeho ekvivalent, 256MB RAM

2.4.2 Požiadavky na softvér

Bohatý klient:
- .NET Framework 1.1
- operačný systém Windows(okrem Windows 95)

Jednoduchý klient:
- internetový prehliadač

Server
- Windows Server 2003
- Internet Information Services(IIS), je súčasťou Windows Server 2003
- SQL Server 2000 (len v prípade ukladania dát do databázy)
- Mobile Internet Toolkit (pre wap stránky)

2.4.3 Časové požiadavky

Keďže predpokladaná záťaž projektu je 120 až 150 človekohodín, tím pre svoju prácu potrebuje minimálne 3 hodiny týždenne strávené v pridelenej výpočtovej miestnosti. Predpokladaná záťaž na jednotlivých členov tímu však môže byť v priebehu vývoja projektu vyššia, najmä vo fáze implementácie.

3 Záver

3.1 Zhodnotenie

V časti návrhu architektúry obaľovača boli uvedené viaceré ponúkané technológie prístupu k problematike. Členovia nášho tímu sú kvalifikovaní pre implementáciu zvolenej technológie na veľmi kvalitnej úrovni s použitím minimálneho množstva zdrojov. Tímoví pracovníci sú pripravení vložiť do zadanej úlohy najvyššiu možnú mieru kreativity a spraviť tak vytvorený obaľovač do istej miery jedinečný v sérii už existujúcich produktov. K tomu by mala prispieť skutočnosť, že sme pripravení implementovať aj generátor WAP stránok z výstupných údajov obaľovača.

Prípadným pokračovaním tohto projektu v budúcnosti by mohlo byť rozšírenie navrhnutého riešenia o možnosť priameho zasielania SMS správ s pracovnými ponukami. Užívateľ by sa na WAPe (resp. Internete) uviedol svoje požiadavky na pracovné ponuky a vyhovujúce ponuky by sa mu istý čas posielali na jeho mobilný telefón. Takéto prípadné pokračovanie tohoto projektu by však vyžadovalo nejakého sponzora. Už pri písaní tejto ponuky sme natrafili na množstvo praktických využití takéhoto wrappera, čo nás ešte viac zaujalo a navnadilo na dokumentovanie našich myšlienok.

3.2 Použité zdroje

[1] Simon Robinson + kolektív: C# Programujeme profesionálně. Computer Press 1130 s. 2003

Obrázky získane z:
- ponuky tímoví projekt č. 10, 2004
- ponuky tímoví projekt č. 4, 2003

Obrázky mobilných telefónov získane z www.t-mobile.sk

Príloha A – Zoradenie tém podľa priority

1. Analýza textov pracovných ponúk z webu
2. Báza znalostí a zručností študentov
3. Portál pracovných príležitostí
4. Tvorba rozvrhov
5. Animácia ľudskej nohy
6. Kandidát na najlepší multimediálny produkt roku 2006
7. Nástroj na modelovanie vlastností
8. RoboCup – nové stratégie
9. Distribuovaná simulácia rozsiahlych počítačových sietí
10. Robocup 3D

Príloha B – Rozvrh členov tímu v zimnom semestri

	7:20 8:10	8:15 9:05	9:15 10:05	10:10 11:00	11:10 12.00	12:05 12:55	13:05 13:55	14:00 14.50	15.00 15.50	15.55 16.45	16.55 17.45	17.50 18.40	18.50 19.45	19.50 20.40
Pon				NS E701			**OOAaNS C802**		**OOAaNS C802**		TimProj DE150			
Uto	***BPS DE150***													
Str							**ZK**		**TK**					
Stv			ASS BC 150				**NS c117a**		MSI BC150		MSI cd 35			
Pia														

NS - Všetci
NS - **Attila Kotrba, Laco Kočiš**
NS - **Lukáš Kročka**
NS - ***Ivan Blanárik***
NS - **Mário Lenický**
\- Najviac preferované termíny (radšej neskoršie večer, ako skoršie)
\- Menej preferované termíny